IMA Journal of Complex Networks (2017) Page 1 of 27
doi:10.1093/comnet/xxx000

Network Analysis using Entropy Component Analysis

CHENG YE
Department of Computer Science,
Royal Holloway, University of London,
Egham, TW20 0EX, UK.
cheng.ye@rhul.ac.uk

AND

RICHARD C. WILSON
Department of Computer Science,
University of York,
York, YO10 5GH, UK.

AND

EDWIN R. HANCOCK
Department of Computer Science,
University of York,
York, YO10 5GH, UK.

[Received on 31 August 2017]

Structural complexity measures have found widespread use in network analysis. For instance, entropy can be used to distinguish between different structures. Recently we have reported an approximate network von Neumann entropy measure, which can be conveniently expressed in terms of the degree configurations associated with the vertices that define the edges in both undirected and directed graphs. However, this analysis was posed at the global level, and did not consider in detail how the entropy is distributed across edges. The aim in this paper is to use our previous analysis to define a new characterization of network structure, which captures the distribution of entropy across the edges of a network. Since our entropy is defined in terms of vertex degree values defining an edge, we can histogram the edge entropy using a multi-dimensional array for both undirected and directed networks. Each edge in a network increments the contents of the appropriate bin in the histogram, indexed according to the degree pair in an undirected graph or the in/out-degree quadruple for a directed graph. We normalize the resulting histograms and vectorize them to give network feature vectors reflecting the distribution of entropy across the edges of the network. By performing principal component analysis (PCA) on the feature vectors for samples, we embed populations of graphs into a low-dimensional space. We explore a number of variants of this method, including using both fixed and adaptive binning over edge vertex degree combinations, using both entropy weighted and raw bin-contents, and using multi-linear principal component analysis (MPCA), aimed at extracting the tensorial structure of high-dimensional data, as an alternative to classical PCA for component analysis. We apply the resulting methods to the problem of graph classification, and compare the results obtained to those obtained using some alternative state-of-the-art methods on real-world data.

Keywords:

von Neumann entropy, entropy component analysis, feature vector

i

1. Introduction

Vertex degree distributions have proved to be a powerful tool in the analysis of complex networks, and allow different network behaviors, such as the Erdős-Rényi, "small-world" and "scale-free" models to be easily distinguished. This analysis can be extended to edges, and leads to the definition of network characteristics, such as the Estrada index [Estrada, 2010] which aims to measure the variance of vertex-degree across edges in a network and allows the homogeneity of networks to be assessed. Such measures prove to be powerful when the analysis of highly inhomogeneous structures such as protein-protein interaction networks is attempted.

The problem of determining the homogeneity of network substructure can also be posed in terms of entropy, and there are a plethora of alternative ways in which network entropy can be measured. For instance, one approach is to use entropy to characterize the degree of randomness in a network [Dehmer et al., 2013]. One of the earliest and classical contributions in randomness complexity is Körner's entropy associated with a graph [Körner, 1973]. The original motivation of this measure is to compute the entropy of an information source with ambiguous alphabet. Extending this idea to graphs, the Körner's entropy is defined as the minimal cross entropy between the vertex packing polytope and the vertex probability distribution [Escolano et al., 2012]. Unfortunately, as this complexity measure is posed as a coding optimization problem in information theory, it cannot be used as a quantity to reflect the graph structural properties. Another drawback of this approach is that it is not applicable to more general large-scale graphs. These shortcomings seriously limit the direct use of Körner's entropy in the field of network analysis.

A number of alternative methods therefore use Shannon's entropy for quantifying the complexity of a graph. Shannon's entropy function can be directly applied to a probability distribution whose values are assigned by functions that capture the structural characteristics of a graph. For instance, Dehmer has proposed a generalized framework for defining graph entropies based on the development of the graph topological information functionals [Dehmer, 2008]. The resulting entropies have proved to be useful in classifying graphs according to structure, and the computational complexity is polynomial. By applying an entropy function to the normalized degree correlation matrix, Claussen et al. have defined an "off-diagonal" complexity measure [Claussen, 2006]. The idea is to define a biased edge distribution entropy, whose extremal value is reached for a power-law distribution. Extending this approach to the distribution of correlations between the degrees of pairs of vertices, a discrete graph entropy can be defined and computed. This complexity takes on the value zero for both regular lattices and complete graphs, and has small values for random graphs and large values for complex structures.

The main drawback of the above methods for randomness complexity is that they do not capture properly the correlations between vertices [Feldman and Crutchfield, 1998]. Statistical complexity aims to overcome this problem by measuring regularities beyond randomness, and does not necessarily grow monotonically with randomness. It is natural to realize that both completely random systems and completely ordered ones should have a minimal statistical complexity. The first randomness complexity measure introduced, namely the Kolmogorov complexity [Kolmogorov, 1998] of an object, is quantified by the length of the shortest algorithm required to reproduce the object. On the other hand, the statistical counterpart of Kolmogorov complexity, the logical depth devised by Bennett [Bennett, 1986], is a measure of complexity based on the algorithmic information and computational complexity of an algorithm which can be used to recreate a given piece of information. In essence, the logical depth complexity measure differs from its randomness counterpart in that it is based on the notion of a process rather than

a measure. Recent additions to the statistical complexity literature are the graph spectral methods. In fact, there exist strong links between the eigenvalue-based and polynomial-based approaches and many practical graph structure characterizations have been developed based on such connections. For example, Luo et al. [Luo et al., 2003] have defined the eigenmodes using the leading eigenvectors of the graph adjacency matrix, and that can be used to compute vectors of spectral properties. Then, graphs can be embedded in a pattern space via those vectors. The method has proved to be efficient in overcoming problems such as graph clustering and object identification. It is also known that the spectrum of the graph Laplacians can be used as an elegant means of characterizing the topological structure of graphs. For instance, Wilson et al. [Wilson et al., 2005] focus on the Laplacian spectral decomposition and show how the coefficients of the permutation invariant polynomials that are computed from the elements of the spectral matrix for the graph Laplacians, can be used as features that capture the metric structure of graphs. Another important example is furnished by Estrada's network heterogeneity index [Estrada, 2010]. In effect, this index gauges differences in degree for all pairs of connected vertices and is dependent on vertex degree statistics and graph size. The expression for the index can be expressed in terms of the Laplacian matrix of graphs. The lower bound of this quantity is zero, which occurs for a regular graph while the upper bound is equal to one, which is obtained for a star graph.

The aim in this paper is to develop a method for exploring in more detail the distribution of entropy over edges in a network, and to characterize the variance in this distribution. We commence from a recently developed von Neumann entropy approximation for a graph, which is based on the degree statistics of vertices connected by edges [Han et al., 2012, Ye et al., 2014]. From this approximate expression, we compute the local von Neumann entropy contribution associated with each edge and calculate the corresponding probability distribution in order to obtain a multi-dimensional entropy histogram. So each edge indexes a histogram bin via its degree configuration. In the case of undirected edges the histogram is two-dimensional, while in the case of directed edges it is four-dimensional. We then normalize the histogram bin-contents and vectorize its contents to extract a feature vector that can be used to effectively represent the statistical distribution of entropy across the edges of an individual network. To overcome problems of storage memory especially in the case of directed graphs, we use an adaptive method to select histogram bins based on quantiles of the cumulative degree distribution. To analyze samples of networks, we apply principal component analysis to a sample of long-vectors, representing the contents of the relevant entropy histogram. This involves first centering the sample of long-vectors so that they have zero mean, and then computing the sample covariance matrix. The eigenvectors of the covariance matrix form the columns of a rotation matrix, and this can be used to rotate the centered long-vectors into the directions of principal covariance. Viewed in this way our method is a form of entropy component analysis (ECA), and closely akin to the method recently reported in [Jenssen, 2010].

The new approach offers a number of advantages over the use of simple vertex degree distributions. First the method relies on a vectorial representation of each network under study, rather than a unary one such as edge density, average degree, degree variance or Estrada index [Estrada, 2010]. This means that each network resides in a high-dimensional feature space, and information concerning its edge structure is not discarded. This potentially improves the separability of samples of networks into different classes when machine learning methods are applied to the vectors. Second, our method makes use of an approximation of the von Neumann entropy expressed in terms of contributions arising from different vertex degree combinations on the edges of a network. The consequences of this are twofold. First, the computation of the von Neumann entropy is expressed in terms of the normalized Laplacian eigenvalues, this not only means that the entropy requires the computation of the Laplacian spectrum (which is usually cubic in the number of vertices in the network) but that the entropy is not resolvable along the edges of

the network. The computation of the approximate entropy on the other hand resolves the entropy on the edges and its computational complexity depends on the number of edges in the network which is at most quadratic in the number of vertices. Second, in common with the Estrada index [Estrada, 2010] it allows us to explore the variance-covariance structure of the degree distribution. However, our method focuses on analyzing the variance in the bin-contents of the edge entropy histogram indexed by degree. Thus if a particular histogram bin, corresponding to the entropy associated with edges having a particular degree combination, has small variance over the sample then it does not generate a significant entropy component in our analysis. On the other hand, those edges associated with large variance in entropy bin-contents, do feature significantly in the component analysis. By performing PCA we identify those degree combinations that give rise to greatest variations in entropy in the sample of graphs under study. When analyzing samples of graphs that change with time or some other systematic variable, it allows those degree combinations which are associated with greatest variation in network entropy to be identified. Moreover, when PCA is applied, it allows networks that have similar patterns of variation in edge degree structure to be grouped together. If on the other hand, MPCA is used, then because it relies on a tensor decomposition of the bin-contents covariance matrix, the variance structure is additionally sensitive to the degree ordering of the bins.

The price to paid for these advantages is the storage required for the histogram of edge entropies. For undirected graphs the storage required is quadratic in the number of vertices in the graph while for directed graphs it is quartic, if binned at full degree resolution (i.e., one bin for each potential edge degree combination). For large graphs, especially in the case of directed graphs, this can prove excessive and also introduces problems with low bin occupation. We overcome this problem by using an adaptive strategy rather than a fixed binning to select the binning. This introduces additional overheads in the pre-processing of network datasets, but reduces the requirements on storage and difficulties associated with low or zero bin-contents. Moreover, when applying adaptive binning to graph entropy distribution histograms, the resulting bin-contents is actually weighted by the entropy of the edges whose degree combinations are similar. However, if we directly apply the same technique to the histograms based on the raw edge degree distribution, then the bin-contents is simply the number of edges having similar degree combinations. Clearly, the former method will give rise to a histogram with a rather different shape.

Hence, it is not the entropy of an edge alone that contributes to the significance of the edge degree combination. It will also be determined by the variance in the number of times such a combination is present in each network in the sample. So although edges connecting vertices of high degree give low values of edge entropy, their significance may be amplified due to the fact that if the community or hub structure changes significantly, then their associated entropy variance will be large. Finally, it should be stressed that the structure of our histogram does not directly relate to the adjacency structure of the network. Bins are in close proximity if they are close in degree, but this does not imply they relate to adjacent vertices or edges. Edges are distributed across different bins of the histogram according to their degree configuration, and not their proximity to one-another.

The remainder of this paper is organized as follows. In Sect. 2 we detail the development of the entropy histogram construction and component analysis method. Section 3 provides an experimental evaluation in order to demonstrate the effectiveness of the proposed method. Finally, we summarize our work and point out a number of future research directions in Sect. 4.

2. Graph Embedding via Edge Entropy Histograms and Component Analysis

We aim to analyze network structure by exploiting the idea of kernel entropy component analysis [Jenssen, 2010]. This is a technique that transforms data to a space spanned by the kernel principal component analysis axes contributing most significantly to the entropy associated with the data. Commencing from an approximation of the von Neumann entropy of a graph, we analyze the entropy contributions originating from each edge. According to our prior work the edge entropies are determined by the degrees of the two vertices defining the edge. Based on this observation we utilize the multivariate distribution of entropy with the different combinations of vertex degree that define edges in a graph. In practice this distribution can be computed by constructing a multi-dimensional histogram whose bins are indexed by the degrees of the connected vertices and whose contents accumulate the edge entropy contributions over the entire graph. The contents of the histogram can be represented by a matrix whose contents can be encoded as a long-vector, and this serves as a feature vector for the graph. There are specific cases for undirected and directed graphs. For undirected graphs the edges are specified by the single degree values for the two participating vertices, and the histogram array is two-dimensional. For directed graphs, on the other hand, there is an in-degree and an out-degree at each vertex participating in an edge, and the histogram array is four-dimensional.

2.1 *Undirected Graphs*

Suppose that $G = (V,E)$ is an undirected graph with vertex set V and edge set $E \subseteq V \times V$, then the adjacency matrix A is defined as follows

$$A_{uv} = \begin{cases} 1 & \text{if } (u,v) \in E \\ 0 & \text{otherwise.} \end{cases}$$

The degree of vertex u is

$$d_u = \sum_{v \in V} A_{uv}.$$

Our prior work on approximating network entropy commences from Passerini and Severini's postulate [Passerini and Severini, 2008] that the combinatorial Laplacian, scaled by the number of vertices in the graph, can be interpreted as the scaled density matrix ρ of an undirected graph. As a result , it is possible to compute the von Neumann entropy $H_{VN} = -Tr[\rho \ln \rho]$ of a graph from the eigenvalues of the associated combinatorial Laplacian. In our analysis, in order to simplify matters we use the normalized Laplacian $\tilde{L} = D^{-1/2}(D-A)D^{-1/2}$ (where D is the degree matrix with the degrees of the vertices of the undirected graph along the diagonal and zeros elsewhere). The choice of normalization is not an important detail since both the Laplacian and normalized Laplacian matrices can be used as valid density matrices. Furthermore, the scaling of the eigenvalues does not affect the functional dependence of the entropy with the degree. In particular, the largest eigenvalue of the Laplacian matrix is bounded by twice the largest vertex degree in a graph, while the normalized Laplacian matrix has eigenvalues between 0 and 2. With this choice of density matrix, the von Neumann entropy of the undirected graph is essentially the Shannon entropy associated with the normalized Laplacian eigenvalues, i.e.,

$$H_{VN}^{U} = -\sum_{i=1}^{|V|} \frac{\tilde{\lambda}_i}{|V|} \ln \frac{\tilde{\lambda}_i}{|V|} \tag{2.1}$$

where $\tilde{\lambda}_i$, $i = 1,2,\cdots,|V|$ are the eigenvalues of the normalized Laplacian matrix $\tilde{L}$.

Commencing from this definition and making use of a second-order Taylor series approximation to the Shannon entropy with expansion point x_0 at the mean value of $\frac{\tilde{\lambda}}{|V|}$, i.e.,

$$x_0 = \frac{\sum_{i=1}^{|V|} \frac{\tilde{\lambda}_i}{|V|}}{|V|} = \frac{Tr[\tilde{L}]}{|V|^2} = \frac{1}{|V|}.$$

The second-order Taylor expansion for $x\ln x$ about the expansion point x_0 is

$$x\ln x = x(\ln x_0 + \frac{x}{2x_0}) - \frac{x_0}{2} + r(x),$$

where $r(x)$ is the remainder term and is computed by

$$r(x) = \sum_{n=3}^{\infty} (-1)^n \cdot \frac{(x-x_0)^n}{x_0^n} \cdot \frac{(n-2)!}{n!}.$$

Note that the eigenvalues $\tilde{\lambda}_i$ of the normalized Laplacian matrix is always bounded between 0 and 2, so we have

$$x - x_0 = \frac{\tilde{\lambda}_i}{|V|} - \frac{1}{|V|} = \frac{\delta}{|V|},$$

where $\delta \in [-1,1]$. This result guarantees that the approximation error of the Taylor series $r(x)$ is a small value since now, we have

$$r(x) = \sum_{n=3}^{\infty} \frac{\delta^n}{n(n-1)}.$$

Substituting this series approximation for the Shannon entropy with expansion point into the expression for the von Neumann entropy Eq. (2.1), we obtain

$$H_T^U = \ln|V| - \frac{1}{2|V|} \sum_{(u,v)\in E} \frac{1}{d_u d_v}. \tag{2.2}$$

This approximation clearly contains two measures of network structure. The first is the number of vertices in the graph $|V|$ while the second is related to the degree statistics of vertices connected by edges. Specifically, the second term of this formula simply sums a degree-based entropy contribution over the edges in a graph. This leads to the possibility of defining a normalized local entropy measure for each individual edge in the graph.

To this end, we normalize this approximate entropy with respect to the total number of edges contained in the graph and thus obtain an expression of the normalized local entropy for the edge (u,v),

$$\gamma_{uv}^U = \frac{\ln|V|}{|E|} - \frac{1}{2|V|d_u d_v}. \tag{2.3}$$

This is precisely the von Neumann entropy contribution of each single edge in the graph, since the sum of these measures over all edges leads to the value of the approximate von Neumann entropy

$$\sum_{(u,v)\in E} \gamma_{uv}^U = \sum_{(u,v)\in E} \frac{\ln|V|}{|E|} - \sum_{(u,v)\in E} \frac{1}{2|V|d_u d_v} = H_T^U.$$

Moreover, this normalized local entropy avoids the graph size bias. In other words, for an arbitrary graph, it is the degree-based edge entropic measure, and not the number of vertices or edges in the graph that distinguishes the entropy contribution from a single edge.

Our entropic histogram construction is based on the statistical information residing in the edges in the graph. In particular, we compute the sum of the normalized local entropy for edges with the same degree configuration, and thus obtain a two-dimensional histogram which represents the edge-based entropy distribution of the graph indexed by degree. To formulate this idea, we first construct an $\alpha \times \alpha$ matrix of zeros M^U (α is the maximum vertex degree of the graph) whose elements are the histogram bin-contents, and where the row and column indices represent the degrees of vertices which run from 1 to α. For instance, the entry M^U_{12} accumulates the entropy contribution for all the edges that connect vertices with degrees 1 and 2.

To compute the bin-contents we proceed as follows. First, we calculate the normalized contribution to the entropy from each edge in Eq. (2.3), then we accumulate the sum over all edges that have the same degree combination. We store this accumulated sum in the corresponding element of the matrix M. The elementwise accumulation is formally given as

$$M^U_{ij} = \sum_{\substack{d_u=i,d_v=j,\\(u,v)\in E}} \left\{ \frac{\ln|V|}{|E|} - \frac{1}{2|V|d_u d_v} \right\}, \tag{2.4}$$

where $i, j = 1, 2, \cdots, \alpha$. It is worth noting that since we consider only undirected graphs here (directed graphs are considered later on), there is no direction information on each edge. As a result the matrix is symmetric since there is no difference between the elements M^U_{ij} and M^U_{ji}. So for convenience, we do not store the elements in the lower triangle below the main diagonal i.e., the matrix M^U is upper triangular. To vectorize the matrix M^U to give us a network feature vector, since all of the entries below the main diagonal of M^U are zeros, we can simply list all the upper triangular elements row by row, with the result that

$$V^U = (M^U_{11}, M^U_{12}, \cdots, M^U_{1m}, M^U_{22}, M^U_{23}, \cdots, M^U_{\alpha\alpha})^T.$$

Clearly, this feature vector has length

$$\alpha + (\alpha-1) + (\alpha-2) + \cdots + 1 = \alpha(\alpha+1)/2.$$

Later on we describe how samples of these vectorized entropy histograms can be analyzed using PCA. This analyzes the variance of the bin-contents, and identifies those edge degree combinations which give rise to the most significant bin-contents, over the sample of networks studied. These correspond to the most significant structural changes in edge degree composition in the network over the sample studied.

There are a plethora of alternatives to PCA. In some of these the order in which elements of the data being studied is crucial. Here our vector encoding disturbs the neighbour proximity of elements in the histogram array. However, PCA still records the pairwise covariance for all pairs of cells, irrespective of their proximity to one-another. With image data or other spatially structured data, it is sometimes important to preserve the spatial proximity relations between elements. One technique for doing this is multi-linear principal component analysis [Lu et al., 2008]. MPCA can be viewed as a generalization of the classical PCA to tensor objects, which captures most of the original tensorial input structure through a multi-linear projection in order to extract features.

To summarize, our analysis of graph structure is based on the bivariate distribution of edge von Neumann entropy with vertex degree for edges in a graph. Moreover, since the von Neumann entropy quantifies the structural complexity of a graph, our proposed feature vector represents statistical information concerning the local structural properties in the graph.

2.2 *Directed Graphs*

We have repeated the above analysis for directed graphs. Suppose that $G = (V,E)$ is a directed graph with vertex set V and edge set $E \subseteq V \times V$, then the adjacency matrix A is defined as follows

$$A_{uv} = \begin{cases} 1 & \text{if } (u,v) \in E \\ 0 & \text{otherwise.} \end{cases}$$

The in-degree and out-degree of vertex u are

$$d_u^{in} = \sum_{v \in V} A_{vu}, \quad d_u^{out} = \sum_{v \in V} A_{uv}.$$

Repeating the analysis above using Chung's definition of the normalized Laplacian of a directed graph [Chung, 2005], we have shown in [Ye et al., 2014] that the approximate von Neumann entropy for a directed graph is,

$$H_{VN}^{D} = \frac{1}{2|V|}\left\{ \sum_{(u,v)\in E} \frac{d_u^{in}}{d_v^{in} d_u^{out^2}} + \sum_{(u,v)\in E_2} \frac{1}{d_u^{out} d_v^{out}} \right\}, \tag{2.5}$$

where $E_2 = \{(u,v) | (u,v) \in E \bigwedge (v,u) \in E\}$ is the set of bidirectional edges. If the cardinality of E_2 is very small ($|E_2| \ll |E|$), i.e., there are few bidirectional edges and the graph is strongly directed (SD), this expression can be simplified one step further by ignoring the summation over E_2, with the result that

$$H_{VN}^{SD} = \frac{1}{2|V|} \sum_{(u,v)\in E} \left\{ \frac{d_u^{in}}{d_v^{in} d_u^{out^2}} \right\}. \tag{2.6}$$

For a directed edge (u,v) the entropy is

$$\gamma_{uv}^{D} = \frac{d_u^{in}}{|V| d_v^{in} d_u^{out^2}}. \tag{2.7}$$

If this edge is bidirectional, i.e., $(u,v) \in E_2$, then we add an additional entropy contribution to γ_{uv}^{D}

$$\Delta\gamma_{uv} = \frac{1}{|V| d_u^{out} d_v^{out}}.$$

This local measure represents the entropy associated with each directed edge since for arbitrary directed graphs, we have $\sum_{(u,v)\in E} \gamma_{uv}^{D} + \sum_{(u,v)\in E_2} \Delta\gamma_{uv} = H_{VN}^{D}$ and for strongly directed graphs, we also have $\sum_{(u,v)\in E} \gamma_{uv}^{D} = H_{VN}^{SD}$. Moreover, this measure avoids the bias caused by graph size. In other words, the edge entropy is determined by the in and out-degree statistics, and not by either the number of vertices or number of edges.

We can again histogram the directed edge entropies. The histogram bins are now indexed by the in and out-degrees of the starting and end vertices. We represent this distribution of entropy using a four-dimensional histogram over the in and out-degrees of the two vertices. Since all the vertices in the graph have in-degree and out-degrees ranging from 1 to β where $\beta = \max(\max(d_u^{in}, d_u^{out}))$, $u = 1,2,\cdots,|V|$, we can then simply construct the directed edge entropy histogram whose size in each dimension is fixed to β. The histogram is stored as a four-dimensional array.

To do this, we first construct a $\beta \times \beta \times \beta \times \beta$ array M^D whose elements represent the histogram bin-contents, and whose indices represent the degrees of the vertices. For instance, the element M_{1234}^{D}

accumulates the entropy contribution for all the directed edges starting from vertices with out-degree 1 and in-degree 2, pointing to vertices with out-degree 3 and in-degree 4. We then compute the bin-contents by summing the directed edge entropy contributions over the sample graph. The histogram bins contain all directed edges having the same degree combinations, so each edge contained in bin contributes an equal amount of entropy. We store the accumulated bin-contents in the corresponding element of array M^D. The elementwise accumulation is formally given as

$$M^D_{ijkl} = \sum_{\substack{d_u^{out}=i,d_u^{in}=j \\ d_v^{out}=k,d_v^{in}=l \\ (u,v)\in E}} \left\{ \frac{d_u^{in}}{|V| d_v^{in} d_u^{out^2}} \right\}. \tag{2.8}$$

If the graph contains bidirectional edges, we additionally accumulate the following quantity

$$\Delta M_{ijkl} = \sum_{\substack{d_u^{out}=i,d_u^{in}=j \\ d_v^{out}=k,d_v^{in}=l \\ (u,v)\in E_2}} \left\{ \frac{1}{|V| d_u^{out} d_v^{out}} \right\},$$

where $i,j,k,l = 1,2,\cdots,\beta$. Repeating the vectorization technique we have introduced for undirected graphs, here we can extract a feature vector from M^D by simply listing all the elements in the array in order, with the result that

$$V^D = (M^D_{1111},\cdots,M^D_{111m},M^D_{1121},\cdots,M^D_{\beta\beta\beta\beta})^T.$$

This feature vector is of length β^4.

It is worth pausing to consider the case of strongly directed graphs. For such graphs, from Eq. (2.6) we note that the directed edge entropy does not depend on d_v^{out}. As a result the dimensionality of the corresponding histogram can be reduced from four to three by ignoring the third index k in M^D_{ijkl}. This leads to a new feature vector with significantly shorter length.

2.3 *Adaptive Histogram Binning*

One of the problems that potentially limits our graph embedding method is that the vertex degree is unbounded. Since our idea relies on capturing the distribution of entropy associated with the edge degree statistics of graphs, then different vertex degree distributions will lead to different distributions of edge entropy. For different graphs being analyzed or compared, these distributions may vary significantly. Moreover, the degree is unbounded. These factors have direct implications on the binning of the edge entropy according to degree. On the one hand we require that the binning of degree is not too fine. If we select too fine a binning, then the storage requirements grow, and this proves to be particularly important in the case of directed graphs, where the histogram is potentially four-dimensional. Moreover, too fine a binning can lead to many empty or sparsely populated bins, thus rendering the representation unstable. On the other hand, if the binning is too coarse then it may fail to capture sufficient fine detail of the distribution of entropy with degree. Finally, it is important that the histograms for different graphs have the same binning so that the entropy distributions with degree are in correspondence and their contents can be meaningfully compared.

We therefore require a histogram binning method that keeps the number of bins fixed so that the vectorized representation of the histogram is of constant length for graphs with a large variance in vertex

degree. Generally speaking, there are two main histogram binning approaches, namely a) fixed binning and b) adaptive binning. On the one hand, in a fixed binning histogram, the distribution of numerical data is partitioned into rectangular bins whose sizes are the same. On the other hand, the adaptive binning approach adapts the bin-size in order to accurately capture the distribution of the data. The main difference between these two approaches is that the former approach requires the same binning to be applied to the entire sample population while the latter allows different binnings for different samples of graphs. It is for this reason that the adaptive binning method is more accurate in representing the true distribution of data than the fixed binning method.

In our analysis, we are dealing with a large number of graph samples and each graph yields an individual entropy distribution histogram. To make the feature vector length constant for all samples, here we adopt the adaptive binning method. To commence, we introduce the cumulative distribution function (CDF) and its quantiles. In particular, we use the vertex degree probability distribution to construct the CDF, from which we can determine the m-quantiles, which divide the ordered vertex degree data into m essentially equal-sized parts. This allows us to relabel each vertex with a specific quantile label $1,2,\cdots,m$. As a result, the number of the bins of the revised histogram is not sensitive to the variance of the degree distribution.

Suppose the undirected graph dataset under study has n vertices in total with an ordered degree sequence $d_1 \leqslant d_2 \leqslant \cdots \leqslant d_n$, then the degree distribution is the probability distribution of these degrees over the entire graph. The corresponding CDF is then given by

$$F_x(d_i) = p(x \leqslant d_i),$$

where $i = 1,2,\cdots,n$. This function describes the probability that a given degree x takes on a number less than or equal to degree d_i.

Quantiles are the samples taken at regular intervals from the CDF of vertex degree distribution. Specifically, they divide the ordered degree data $d_1,d_2,\cdots,d_n$ into a number of equal-sized data subsets. Since the vertex degree is always a non-negative integer, the quantiles of the CDF can be viewed as new vertex labels which represent intervals spanned by the degree. In our analysis, we let the number of subsets be m, so the m-quantiles can be obtained as follows

$$Q_j^U = \underset{d_i}{\operatorname{argmin}} \left\{ F_{Q_j^U}(d_i) - \frac{j}{m} \right\}, \tag{2.9}$$

where $i = 1,2,\cdots,n$ and $j = 1,2,\cdots,m$. These new degrees satisfy $Q_1^U \leqslant Q_2^U \leqslant \cdots \leqslant Q_m^U$ and in fact, $Q_m^U = d_n$.

With these ingredients, we can give each vertex in the graph dataset a new label indexed by the quantile into which the original degree falls. To do this, for a given vertex u, we first examine its original degree d_u, if d_u satisfies the condition that $Q_{k-1}^U < d_u \leqslant Q_k^U$, then vertex u is labeled with quantile label $q_u = k$ (here we define $Q_0^U = 0$). With all the vertices in the graph having quantile labels from 1 to m, we then can simply construct a two-dimensional edge-based entropy histogram whose bin number in each dimension is fixed to m.

Next we repeat the development of the entropic histogramming method proposed in the previous subsection. For each undirected graph $G = (V,E)$ in the dataset, we again construct an $m \times m$ zero matrix W^U where the row and column indices represent the new degree labels of vertices and run from

1 to m. The elementwise computation for the matrix is formally given as

$$W_{ij}^{U} = \sum_{\substack{q_u=i,q_v=j,\\(u,v)\in E}} \left\{ \frac{\ln|V|}{|E|} - \frac{1}{2|V|d_u d_v} \right\}, \tag{2.10}$$

where $i,j = 1,2,\cdots,m$ and $q_u = i$ denotes that vertex u, with original degree d_u is assigned a new degree label i.

A similar analysis can be applied to directed graphs. Suppose a set of directed graphs has n vertices in total which have been sorted according to in-degree (or out-degree) in the sequence $d_1^{in} \leqslant d_2^{in} \leqslant \cdots \leqslant d_n^{in}$. Let $p(x = d_i^{in})$ be the in-degree probability distribution of the graph. The corresponding CDF for the in-degree is given by

$$F_x(d_i^{in}) = p(x \leqslant d_i^{in}),$$

where $i = 1,2,\cdots,n$. This function describes the probability that a given in-degree x takes on a number less than or equal to d_i^{in}.

As discussed previously, the quantiles divide the ordered data $d_1^{in}, d_2^{in}, \cdots, d_n^{in}$ into a number of equal-sized data subsets. Since vertex degree is always a non-negative integer, the quantiles can thus be viewed as new quantization of the degree based on its statistical distribution. We define the degree quantiles over the cumulative distribution of degree for the entire sample of graphs under study, and produce requantized versions of the individual entropy histograms for each individual graph. Suppose the number of quantiles in each dimension of the degree distribution is fixed to be m. Then, for example, the m-quantiles of the in-degree distribution can be obtained as follows

$$Q_j^D = \underset{d_i^{in}}{\operatorname{argmin}} \left\{ F_{Q_j^D}(d_i^{in}) - \frac{j}{m} \right\}, \tag{2.11}$$

where $i = 1,2,\cdots,n$ and $j = 1,2,\cdots,m$. Clearly, these degree quantiles satisfy $Q_1^D \leqslant Q_2^D \leqslant \cdots \leqslant Q_m^D$ and in fact, $Q_m^D = d_n^{in}$.

With the sample degree quantiles to hand, we assign each vertex degree quantile labels. We first examine the original in-degree d_u^{in} of a vertex u, if d_u^{in} satisfies the condition that $Q_{k-1}^D < d_u^{in} \leqslant Q_k^D$ (here $Q_0^D = 0$), then its in-degree quantile is $q_u^{in} = k$. The corresponding out-degree quantile labels can also be obtained in the same manner.

Then, given a directed graph $G = (V,E)$, all the vertices in the graph have in-degree and out-degree quantile labels ranging from 1 to m. This allows us to construct the directed edge entropy histogram whose size in each dimension is fixed to m. The histogram is stored as a four-dimensional array.

To do this, we first construct an $m \times m \times m \times m$ array W^D for each sample of graph dataset, and whose element is given as

$$W_{ijkl}^{D} = \sum_{\substack{q_u^{out}=i,q_u^{in}=j\\q_v^{out}=k,q_v^{in}=l\\(u,v)\in E}} \left\{ \frac{d_u^{in}}{|V|d_v^{in}d_u^{out^2}} \right\}. \tag{2.12}$$

If the graph contains bidirectional edges, we additionally accumulate the following quantity

$$\Delta W_{ijkl} = \sum_{\substack{q_u^{out}=i,q_u^{in}=j\\q_v^{out}=k,q_v^{in}=l\\(u,v)\in E_2}} \left\{ \frac{1}{|V|d_u^{out}d_v^{out}} \right\},$$

where $i, j, k, l = 1, 2, \cdots, m$.

When accumulated in this way we effectively count directed edges with the same configurations of degree quantile labels, and weight them according to their entropy. If the different quantile labels were independent, we would expect a uniform histogram. However, structure in the individual sample graphs due to preferred combinations of vertex in-degree and out-degree will give rise to a non-uniform distribution. To some extent, the quantization of the distribution of entropy with degree according to quantile labels, may dilute this structure due to merging adjacent degree bins. However, the directed edge entropy contribution is based on the original vertex in and out-degree statistics, and the m-quantiles play a role in diminishing the bias caused by different populations of directed graphs. Therefore the entropic representation can still be effective in capturing statistical information concerning the local structural properties in the graph. By embedding graphs into a space spanned by feature vectors, it provides a theoretically principled and efficient tool for graph characterization tasks, which captures the graph characteristics at both the statistical and structural levels.

2.4 *Principal Component Analysis on Vectorized Entropy Histograms*

The main aim in this paper is to compare network structures using the vectorized contents of the edge entropy histograms. To this end, we proceed as follows. Given a sample or dataset of graphs (undirected or directed), we first construct the histogram using the degree quantiles if necessary, for each graph and then vectorize it. We first extract the entire set of vertex degrees from it. Then, we apply the quantization to this vertex degree sequence in order to obtain the quantile values. This in turn allows us to construct a multi-dimensional histogram in which each dimension has the same number of bins for each individual graph from the sample. We then perform PCA on the feature vectors extracted from the histograms in order to analyze the entropy distribution, and also to visualize and compare the sample of graphs in a low-dimensional feature space.

Suppose that $K = \{X_1, \cdots, X_N\}$ is a set of vectorized histograms extracted from the sample of N networks. The mean vector is

$$\hat{X} = \frac{1}{N}\sum_{i=1}^{N} X_i.$$

The centered long-vectors are obtained by subtracting the mean, and so the i-th centered long-vector is given by $\tilde{X}_i = X_i - \hat{X}$. The centered long-vectors are used as columns of the data matrix $K = (\tilde{X}_1|\cdots|\tilde{X}_N)$, and the sample covariance matrix is

$$\Sigma = \frac{1}{N} K K^T.$$

To perform PCA, we compute the eigen-decomposition

$$\Sigma = Y \Lambda Y^T$$

where Λ is the diagonal matrix with ordered eigenvalues as entries, and Y is the matrix with the correspondingly ordered eigenvectors as columns. The centered long-vectors can be transformed into the co-ordinate system aligned with the directions of principal variance using the rotation

$$X_i' = Y \tilde{X}_i.$$

To project the transformed data into a low-dimensional space spanned by the entropy components, we simply select the required number of leading components of the vector X_i'. These components correspond to the linear combinations of histogram bins of greatest bin-contents variance, i.e., entropy

variance. These bins therefore reveal those degree combinations whose frequency varies most over the sample of graphs studied. In other words, the entropy component analysis reveals those combinations of bins that give the greatest spread of networks in the sample. Since the bin-contents is determined by the frequency of the different degree combinations times the corresponding edge entropy, the effect of the entropy is to weight the effects of bin-contents. Edges connecting low degree vertices will therefore have a greater significance than those connecting high degree vertices.

It is worth pausing to discuss the role of the quantization method used. Here we perform the adaptation of the vertex degrees globally, i.e., the quantile values are obtained from the entire sequence of vertex degree values of graphs. This is because such a method allows the graph structures to be compared over the same intervals of degrees. If, on the other hand, we do not estimate the quantiles for the entire sample, then the different entropy histograms are not in correspondence with respect to degree and their comparison is meaningless.

3. Experiments

We have described a graph characterization based on the distribution of entropy over the edges of directed and undirected graphs. To evaluate this novel method and analyze its properties, in this section we employ the method to solve a number of graph classification problems and compare the performance with that of several state-of-the-art techniques. In particular, we perform PCA over a set of histogram vectors extracted from graphs. This allows us to identify a new basis associated with maximum variation in entropy. This is an effective form of ECA since the directions of maximum entropy variation correspond to those where there is maximum variation in degree structure for the edges.

We first provide a brief overview of the datasets used, which include both undirected and directed graphs.

- *ENZYMES*. Is a dataset of protein tertiary structures obtained from the Protein Data Bank [Berwanger et al., 2012]. It consists of 600 enzymes, which are extracted from the BRENDA enzyme database [Schomburg et al., 2004]. Each graph belongs to one of six Enzyme Commission top level classes (EC classes). All graphs are undirected and unweighted.

- *MUTAG*. Consists of 188 graphs representing mutagenic aromatic and heteroaromatic nitro compounds. They are assigned to two classes according to whether or not they have a mutagenic effect on the Gram-negative bacterium *Salmonella typhimurium*. All graphs are undirected and unweighted.

- *NCI1 and NCI109*. Are two balanced subsets of the National Cancer Institute (NCI) database, consisting of graphs representing chemical compounds screened for activity against non-small cell lung cancer and ovarian cancer cell line respectively. The graphs in each subset are labeled as active or inactive. All graphs are undirected and unweighted.

- *D&D*. Contains 1178 proteins, with 691 enzymes and 487 non-enzymes. Each protein is represented by a graph structure, in which the vertices are amino acids while the edges are the connections between amino acids that are less than 6 Ångström apart. All graphs are undirected and unweighted.

- *COIL*. Is a 3D object database constructed by Nene et al. [Nene et al., 1996]. For each object in this 20-object database, 72 images have been collected from equally spaced changes in viewing direction over 360 degrees. For each image, we establish a 6-nearest neighbor graph on the extracted feature points, i.e., each feature point has three directed edges going to its nearest neighbor points, thus the graph is directed and the out-degree of all vertices is 6.

- *NYSE Network*. Is extracted from a database consisting of the daily prices of 3799 stocks traded on New York Stock Exchange (NYSE). This data has been well analyzed in [Silva et al., 2015], which has provided an empirical investigation studying the role of communities in the structure of the inferred NYSE stock market. The authors have also defined a community-based model to represent the topological variations of the market during financial crises. Here we make use of a similar representation of the financial database. Specifically, we employ the correlation-based network to represent the structure of the stock market since many meaningful economic insights can be extracted from the stock correlation matrices [Battiston and Caldarelli, 2013, Bonanno et al., 2004, Caldarelli et al., 2004]. To construct the dynamic network, 347 stocks that have historical data from January 1986 to February 2011 are selected [Peron and Rodrigues, 2011, Silva et al., 2015]. Then, we use a time window of 20 days and slide this window in order to obtain a sequence (from day 20 to day 6004) in which each temporal window contains a time-series of the daily return values of all stocks over a 20-day period. We represent trades between different stocks as a network. For each time window, we compute the cross-correlation coefficients between the time-series for each pair of stocks, and create connections between them if the maximum absolute value of the correlation coefficient is among the highest 5% of the total cross correlation coefficients. This yields a time-varying stock market network with a fixed number of 347 vertices and varying edge structure for each of 5985 trading days. All graphs are undirected and unweighted.

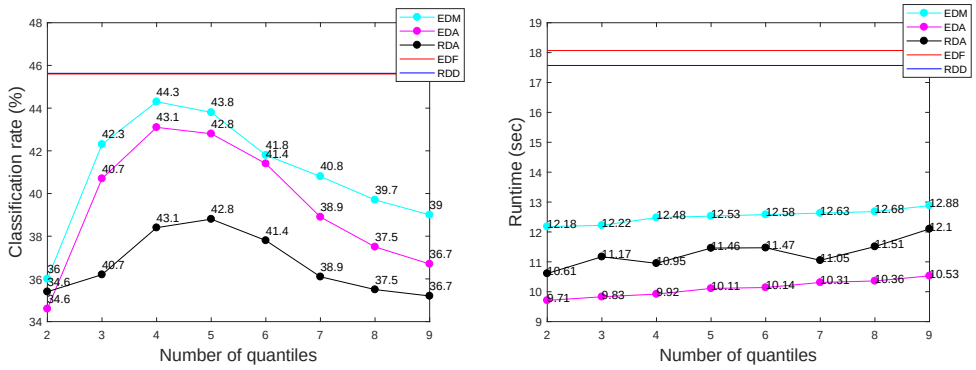
We experiment with a number of variants of the method described earlier in this paper. These involve a) fixed and adaptive binning, b) PCA and MPCA and c) using raw and entropy weighted histograms. In the remainder of the paper, we use the abbreviations a) EDF to denote the use of the entropy distribution obtained from fixed binning with variance component analysis performed using classical PCA, b) EDA to denote adaptive binning and the use of classical PCA and c) EDM to denote adaptive binning and variance component analysis using MPCA. We also explore the use of the raw degree distribution, referred to as RDD when used with fixed binning and RDA when used in conjunction with adaptive binning. Here the histogram bin-contents is incremented by unity for each edge degree combination, rather than weighting the contents according to the associated edge entropy. Both RDD and RDA use classical PCA for variance component analysis.

3.1 *Undirected Graphs*

3.1.1 *Adaptive Histogram Binning*

To commence, we provide a comparison of entropy histograms constructed using a) fixed binning where the bin-size is completely dependent on the span of the vertex degree and b) adaptive binning where the bin-size is determined by the cumulative vertex degree distribution for the entire sample studied. For the latter method, we vary the number of quantiles m associated with the adaptive binning.

For all five methods (three variants of entropy component analysis and two variants of raw degree



(a) Classification accuracy (b) Experimental runtime

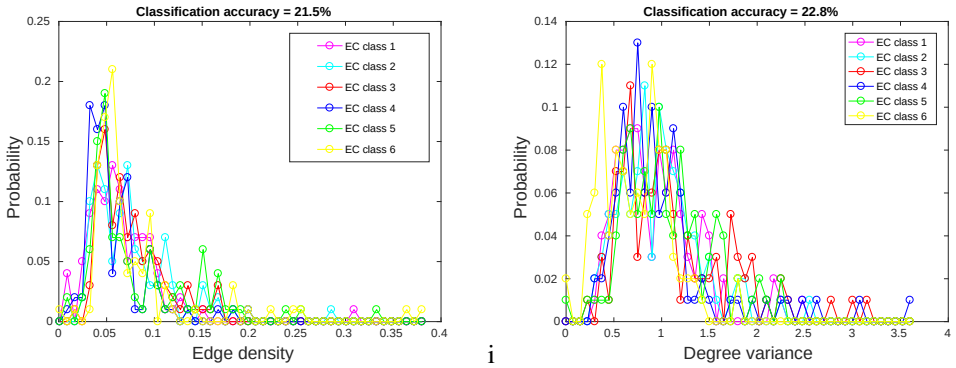
FIG. 1. A comparison of classification performance between EDF, EDA, EDM, RDD and RDA methods.

histogram analysis), we extract the relevant histograms from the different samples of graphs contained in the *ENZYMES* dataset and obtain their corresponding variance components (or feature vectors). After using PCA or MPCA to extract the most significant variance components, we can then employ standard machine learning classifiers to determine the class labels for each graph in the database. In order to have a metric that quantifies the labelling performance, we report the average classification rates of 10 runs together with the computational time of both fixed binning and adaptive binning methods.

Figure 1(a) shows that EDF gives relatively good classification performance, with accuracy equal to 45.6%. This result turns out to be significantly better than a number of state-of-the-art pattern recognition methods on the same data. For instance, the accuracy of the random walk kernel, graphlet count kernel and shortest path kernel are 20.9%, 32.7% and 41.4% respectively. This observation demonstrates the potential of the raw entropy histogram as a representation of graph structure. Also, the classification accuracy of the RDD method is very similar to that of EDF, implying that the when using fixed binning, the histograms arising from raw degree distribution and entropy distribution although different in shape do not give significant performance difference.

Turning attention to the adaptive binning methods EDA, EDM and RDA, the accuracy fluctuates significantly and is heavily dependent on the number of quantiles selected, and whether these provide sufficient resolution to distinguish the structure of the histograms. The maximum number of quantiles in this experiment is set to be 9 as this is the maximum degree value in the graph dataset. In particular, the accuracy of the EDA and EDM methods peak (at 43.1% and 44.3% respectively) when the number of quantiles is 4 and then gradually decreases. Smaller or larger values result in binnings that are either too coarse or too fine, both of which lead to inaccuracy in the classification performance. Thus the entropic histogramming and component analysis algorithm is sensitive to the choice of the appropriate quantile number when performing adaptive histogram binning. Another important feature to note here is that the accuracy of adaptive binning is always lower than that of the fixed binning, regardless of the choice of the quantile number. This is not surprising since the entropy histogram based on raw degree distribution clearly captures more information about the connectivity structure of a graph.

In Fig. 1(b) we compare the computational runtime of the methods used in the above analysis, which includes both feature extraction and model training time. There are a number of important points



(a) Edge density (b) Average degree variance

FIG. 2. Classification performance of graph edge density and average degree variance.

here. First, the runtime of both adaptive binning methods grows with increasing quantile number. This is expected since the size of the entropy histogram is quadratic in the quantile number. Second, the entropy component analysis method is computationally tractable as the runtime does not increase rapidly even when the size of the histogram becomes particularly large. Third, both the EDF and RDD methods take a longer time than EDA, EDM or RDA. This is because the size of the long-vectors extracted via both fixed binning methods are significantly larger than that of the adaptive binning, which requires the classifier to take a significantly longer time to train.

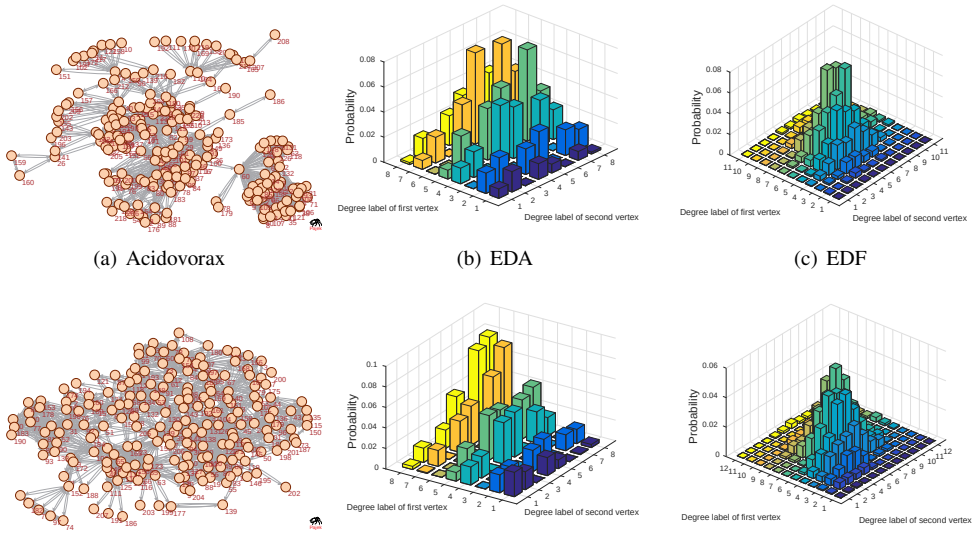
Overall, Figs. 1(a) and 1(b) suggest that compared with the fixed binning method, when the appropriate number of quantiles is used, adaptive binning in conjunction with PCA provides good classification performance in terms of both run time and accuracy. By incorporating MPCA, the classification accuracy can be improved. However, the computational complexity also increases significantly.

In order to investigate whether some simple graph degree statistics can be employed to separate graphs in this dataset, we show the normalized histogram of graph edge density in Fig. 2(a) and average degree variance for the samples in Fig. 2(b) for the *ENZYMES* dataset. In both figures, the histograms of the 6 classes overlap significantly. In fact, the classification rates are only slightly over 21%. This means that entropic histogramming and component analysis method offers a more effective way to capture the structural information when the structure of graphs cannot be well captured by basic unary graph features.

3.1.2 *Graph-based Pattern Recognition*

We have applied the entropy component analysis methods to real-world bioinformatics data. We commence by showing some examples of the entropy histograms obtained from some example networks in the *ENZYMES* dataset. Figures 3(a) and 3(d) show the 2D network representation of two proteins from Acidovorax and Aquifex respectively, while Figs. 3(b), 3(c), 3(e) and 3(f) show their corresponding edge entropy histograms obtained via the EDA and EDF methods. Here the quantile number of the adaptive binning method is set to be 8.

The different networks give rise to significantly different entropy histograms. In particular, we



(a) Acidovorax (b) EDA (c) EDF

(d) Aquifex (e) EDA (f) EDF

FIG. 3. Network representation and edge entropy histogram of two proteins.

Datasets	MUTAG	NCI1	NCI109	D&D	ENZYMES	COIL
Maximal vertices #	28	111	111	5748	126	167
Minimal vertices #	10	3	4	30	2	26
Average vertices #	17.9	29.9	29.7	284.3	32.6	70.8
Graph #	188	4110	4127	1178	600	1440

Table 1. Detailed information of the real-world graph database used in the experiments.

note that the vertex degree of Aquifex network is more uniformly distributed than that of Acidovorax network, which is well reflected in the histogram plots of EDA. Moreover, the degree assortativity properties can also be well captured by the edge entropy histograms, i.e., the low-degree vertices in Aquifex have an approximately equal probability to connect to both high and low-degree vertices while in Acidovorax, the high-degree vertices are more likely to be connected.

Next we compare the classification performance of our entropy component analysis method and its variants with a number of alternative state-of-the-art approaches proposed in the recent literature. In Table 1 we give some statistical information for the datasets used in the experiment, including the number of graphs, the maximal and minimal number of vertices in a graph.

For those methods listed that use adaptive binning, i.e., EDA, EDM and RDA, we seek the number of quantiles m that gives the best performance (normally, m is a small number). We show the comparison results between our methods and several alternative state-of-the-art pattern recognition methods (listed in Table 2). These methods include three feature-vector-based graph embedding methods, namely a) the coefficient feature vector from the Ihara zeta function for graphs (CI) [Ren et al., 2011], b) graph features from topological and label attributes (GF) [Li et al., 2012], and c) the discriminative prototype selection embedding method (DP) [Borzeshi et al., 2013] and three graph kernel methods, namely a) the random walk graph kernel (RW) [Kashima et al., 2003], b) the Weisfeiler-Lehman subtree graph kernel (WL)

Abbreviations	Methods
CI	coefficient feature vector from the Ihara zeta function for graphs
GF	graph features from topological and label attributes
DP	discriminative prototype selection embedding method
RW	random walk graph kernel
WL	Weisfeiler-Lehman subtree graph kernel
GC	graphlet count graph kernel
EDF	entropy distribution with fixed binning and principal component analysis
EDA	entropy distribution with adaptive binning and principal component analysis
EDM	entropy distribution with adaptive binning and multi-linear principal component analysis
RDD	raw degree distribution with fixed binning and principal component analysis
RDA	raw degree distribution with adaptive binning and principal component analysis

Table 2. Abbreviations for the pattern recognition techniques used in the experiment.

[Shervashidze et al., 2010] and c) the graphlet count graph kernel (GC) [Shervashidze et al., 2009].

For the graph embedding methods CI, GF, DP, EDF, EDA, EDM, RDD and RDA we perform either PCA or MPCA to obtain graph features. On the other hand, for the graph kernel methods RW, WL and GC, we first compute their kernel matrix, then perform kernel PCA on this matrix in order to embed the associated data into a principal component feature space. This allows us to employ a diverse array of standard machine learning algorithms for graph classification.

In our comparison, we perform the 10-fold cross-validation using a support vector machine (SVM) classifier associated with the sequential minimal optimization (SMO) [Platt, 1999] and the Pearson VII universal kernel (PUK). All the experiments are performed on an Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz processor, with 8 GB memory. We report the average classification accuracy for each method over the 10-fold cross-validation run 10 times. We also give an evaluation of the runtime of each method, which includes feature extraction time and classifier model training time. It is important to stress that in order to make the comparison fair, for our entropy distribution methods in conjunction with the adaptive binning strategy, i.e, EDA and EDM, we include the parameter-searching (quantile number) time in the overall experimental runtime. Table 3 gives the classification accuracies as percentages while in Table 4, the runtime is given in seconds, minutes and hours. In both tables, "DNF" in a cell indicates that the computation did not finish within a sufficiently long period of time (12 hours, in this experiment). In this case due to the large computational complexity, the experimental run is aborted.

Table 3 shows the classification accuracy comparison for our proposed method and its variants (EDF, EDA and EDM) versus a number of alternative graph embedding methods and graph kernel methods. The corresponding runtime results are shown in Table 4. Compared to the graph embedding methods (CI, GF and DP), the entropic histogramming and component analysis methods give the best classification performance. Specifically, on the *MUTAG* dataset, EDM gives the highest classification rate. On the *NCI1 and NCI109* dataset and *D&D* dataset, our method and its variants (EDF, EDA and EDM) also give the highest classification rates.

For the graph kernel methods, although the classification rates of WL on the *NCI1 and NCI109* dataset are high, our entropic histogramming and component analysis methods still give competitive performance. Another interesting feature in the table is that the RDA method, which uses the raw degree distribution in conjunction with the adaptive binning strategy, shows worse classification results than alternative adaptive-binning-based methods. This is because in RDA, the bin-contents simply counts the

Datasets	MUTAG	NCI1	NCI109	D&D
CI	80.85	60.05	62.79	DNF
GF	86.57	65.81	65.30	69.92
DP	75.61	60.93	60.23	63.19
RW	81.01	DNF	DNF	DNF
WL	84.57	73.00	**73.28**	75.63
GC	84.04	67.71	67.32	77.33
EDF	85.17	72.85	72.87	77.42
EDA	87.83	**73.55**	72.56	75.69
EDM	**88.13**	72.93	72.48	77.28
RDD	86.84	72.92	72.86	**77.94**
RDA	87.51	67.77	68.17	73.16

Table 3. Comparison of graph classification results on bioinformatics graph database (accuracy unit is %).

Datasets	MUTAG	NCI1	NCI109	D&D
CI	0.8 s	45.5 s	47.0 s	DNF
GF	0.6 s	1.2 m	1.2 m	1.1 h
DP	0.8 s	2.9 m	2.8 m	1.6 h
RW	15 s	DNF	DNF	DNF
WL	3.2 s	3.8 m	3.7 m	**17.8 m**
GC	4.6 s	2.2 m	2.2 m	31.5 m
EDF	**0.5 s**	34.5 s	33.9 s	58.1 m
EDA	4.1 s	**31.6 s**	32.0 s	36.9 m
EDM	7.3 s	42.9 s	45.1 s	46.3 m
RDD	**0.5 s**	35.6 s	35.5 s	59.3 m
RDA	3.7 s	32.9 s	**31.5 s**	44.1 m

Table 4. Comparison of experimental runtime on bioinformatics graph database.

number of edges connecting vertices with particular degree combinations. As a result it does not take into consideration the degree weighting associated with edge entropy. On the other hand, EDA ad EDM are able to incorporate such structural information, which turns out to be important for distinguishing different graph structures.

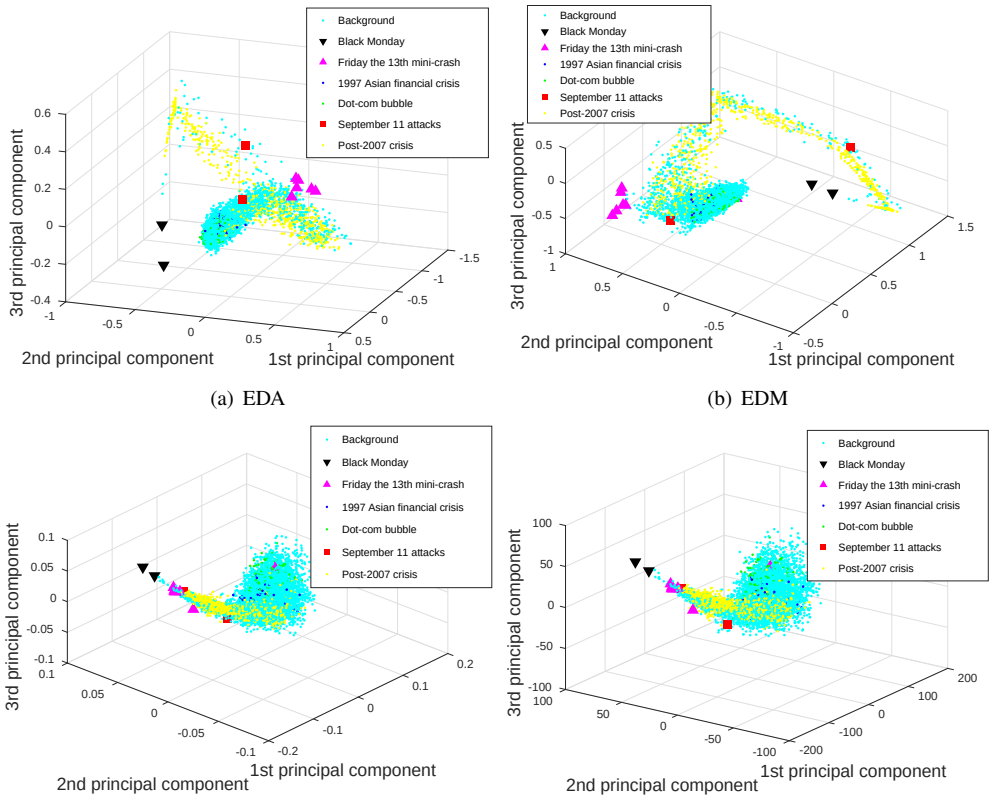
From Table 4, the graph embedding methods based on PCA have faster runtimes than the graph kernel methods. Another interesting feature is that two of the methods did not finish computation within a sufficiently long time on the *D&D* dataset, this is mainly because some graphs in this dataset have more than 5000 vertices (according to Table 1). As a result, the graph embedding and graph kernel methods which rely on the computation over the number of vertices and edges have a significant computational complexity, leading to large runtimes. However, our proposed methods EDA and EDM show good runtime performance as they finished the experimental computation for all four of the datasets with a generally lower runtime. In particular, the entropy distribution and component analysis method EDA has the lowest runtime on the *NCI1 and NCI109* dataset, even when we include the parameter-searching time. Another interesting observation is that on small graphs (*MUTAG* dataset and *NCI1 and NCI109* dataset), the runtimes of EDF and RDD are significantly smaller than those of EDA and EDM. However, when dealing with large graphs (*D&D* dataset), their runtimes become significantly longer, which is caused by the large vertex degrees in graphs and thus large histograms which give rise to long feature vectors.

From this experimental evaluation, our entropic histogramming and component analysis methods have proved to outperform some of the state-of-the-art methods, as they give both accurate and computationally efficient graph classification performance.

3.1.3 *Time Series Structure*

The final experimental goal in this subsection is to investigate whether the entropic histogramming and component analysis technique can be used to detect significant events and to distinguish different periods or epochs in the time evolution of network structure. To this end, we apply all three variants of our method (EDA, EDM and EDF) and the raw degree distribution histogram (RDD) to the graph time-series in the *NYSE Network* dataset and explore how the networks are distributed in the low-dimensional entropy component space. Figure 4 shows the behavior of the leading three principal components. The most interesting feature in Figs. 4(a) for EDA and 4(b) for EDM is that a number of significant outliers correspond to significant financial crises, including the Black Monday and Friday the 13th mini-crash. Moreover, the points respectively representing data before and after September 11 attacks are quite distinct from each other. This suggests that the network structure undergoes a significant change during this period. The figure also shows that the network time-series data has a strong manifold structure with different financial periods occupying different compact volumes in the entropy component space. This observation implies how the network structure evolves coherently over time, but undergoes significant changes in structure at different financial crises. However, in Figs. 4(c) and 4(d), the same features are not easily observed. For example, a number of different financial periods (1997 Asian financial crisis and Dot-com bubble) overlap significantly. This implies that the raw degree distribution fails to preserve some important features which can be used to distinguish graph structures that belong to different classes.

To explore the origin of the variance structure of this data in more detail, we analyze how the contents in each bin in the two-dimensional entropy distribution histogram obtained using the EDA method changes over time. To this end in Fig. 5 we plot the variance of each bin over the entire time-



(a) EDA (b) EDM

(c) EDF (d) RDD

FIG. 4. Scatter plots of the time-evolving financial network in the 3D principal component space constructed from EDA, EDM, EDF and RDD.

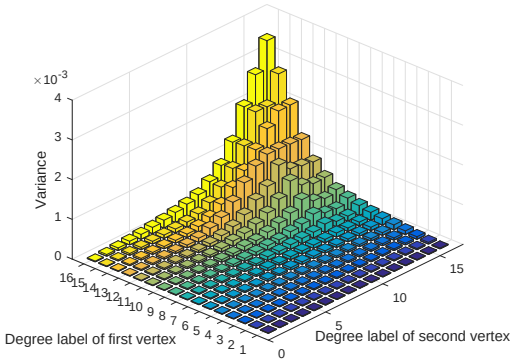
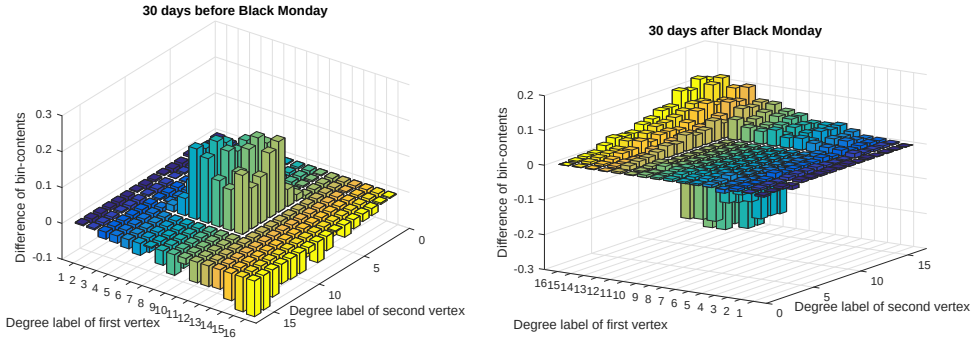


FIG. 5. Bin-contents variance of the time-evolving financial network.



(a) Before crisis (b) After crisis

FIG. 6. Bin-contents difference 30 days before and 30 days after Black Monday.

series. Here the maximum degree label (i.e., the number of quantiles) is set to be 16. The figure shows that the bin-contents variance increases with degree.

Next, we explore in detail how the bin-contents change during periods before and after a particular financial crisis, in this case Black Monday. In Figs 6 we show a) the difference of the bin-contents of the two-dimensional entropy histogram between the close of 30 trading days before Black Monday and Black Monday itself, and b) a similar plot for the difference between Black Monday and the close of 30 trading days after Black Monday. The most significant feature to note is that in the period before Black Monday the population of those bins associated with edges connecting mid-range degrees (i.e., 5 to 10) increases significantly. On the other hand those bins defined by large vertex degrees (i.e., 11 to 16) show a clear drop. After the financial crisis, the bin-contents difference shows the opposite behavior. The two plots together give an intuitive insight into how the entropy distribution of the stock network evolves during a financial crisis.

In Fig. 7 we show the bin-contents of the top five bins with the largest variance as a function of time during the Black Monday crisis for both the EDA and EDM methods. Also shown in this figure is the time-series of the first five principal components which are obtained by applying PCA and MPCA to the feature vectors of the financial network data. Clearly, both the bin-contents and principal components display a significant fluctuation during the crisis, demonstrating again that the entropic histogramming and component analysis method is useful in revealing the structural changes in network evolution. More interestingly, the bins with maximum variance and the most significant principal components show the similar behavior. Specifically, in Fig. 7(b) the lines representing largest variance and principal components almost overlap perfectly, which conforms to the expectation since MPCA identifies components that contribute most to the variance of the data.

This suggests that the bins with largest variance, i.e., the edge configurations that are both frequently occurring and which are defined by large degree combinations, play a dominant role in the PCA. This effect is expected to be the case for edges that constitute large communities or hubs, and these are generally considered as the most important structures in a network. This underlines the utility of our entropy component analysis method for identifying those edge degree configurations responsible for variations in network structure, and those networks where the variation is greatest.

3.2 *Directed Graphs*

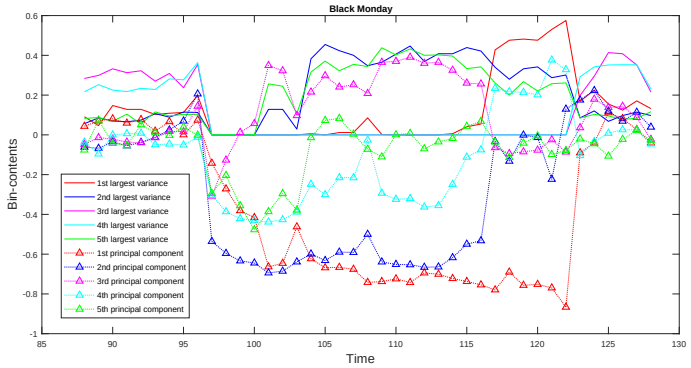
We now turn our attention to directed graphs. We apply entropic histogramming and component analysis method to some real-world data and report the classification results obtained on both *ENZYMES* and *COIL* databases. Here the undirected graphs in *ENZYMES* dataset are converted into directed graphs by constructing a 3-nearest neighbor graph for protein secondary structure elements. In the following evaluation, we perform the 10-fold cross-validation using support vector machine (SVM) classifier associated with the sequential minimal optimization (SMO) [Platt, 1999] and the Pearson VII universal kernel (PUK).

In Fig. 8(a) we report the average classification rates of 10-fold cross validation as a function of number of quantiles m for both EDA and EDM methods on the *COIL* dataset and *ENZYMES* dataset. Figure 8(b) gives the average runtime as a function of the number of quantiles for the experiments on these datasets.

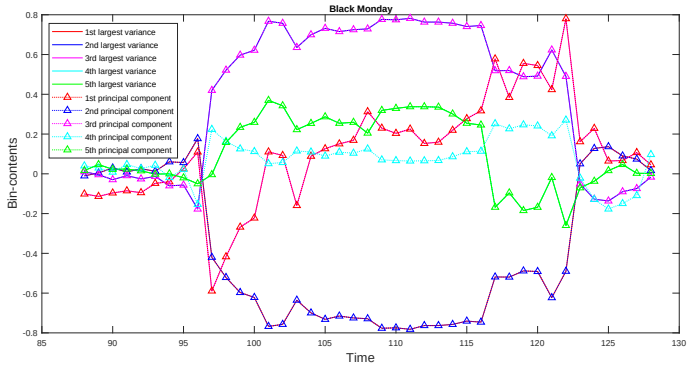
From Fig. 8(b), the experimental runtime for both classification problems grows as the number of quantiles increases, which is as expected since the greater the number of quantiles the greater the size of the histogram and hence the feature vector, resulting in greater computational complexity. Moreover, incorporating MPCA clearly increases the computation time beyond that required by using PCA.

Turning our attention to the classification results reported in Fig. 8(a), the accuracy of EDA on the *COIL* dataset is particularly low. This is not surprising as the classification problem we are dealing with is extremely difficult, i.e., classify objects into 20 groups. However, the performance of the EDM method witnesses a dramatic increase when the quantile number increases from 2 to 3, and the classification rate finally reaches nearly 70%, which is an excellent result.

On the *ENZYMES* dataset, as the number of quantiles increases, the classification rates for both methods show a slight increase, reaching a peak when the number of quantiles reaches 3. Subsequently the performance drops significantly. This is because for the graphs of these datasets, all vertices have the same out-degree 3. Therefore when $m = 3$ the corresponding feature vectors precisely preserve the information of the vertex in and out-degree statistics, which guarantees that $m = 3$ gives the best

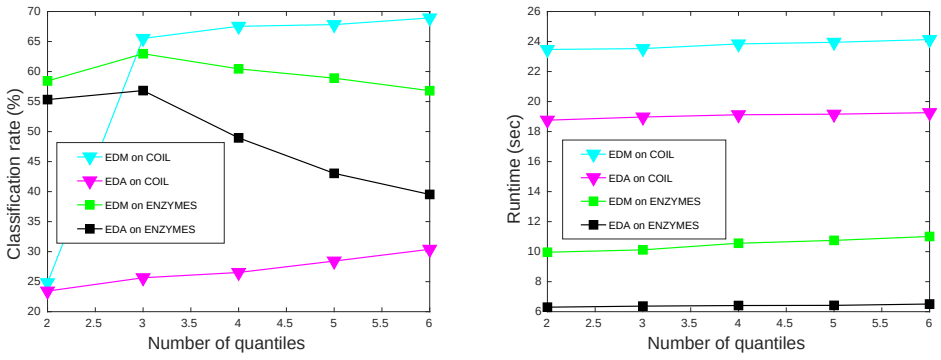


(a) EDA



(b) EDM

FIG. 7. Bin-contents variance of the time-evolving financial network.



(a) Classification accuracy (b) Experimental runtime

FIG. 8. Average classification rates and experimental runtime of EDA and EDM on different datasets as a function of number of quantiles.

classification performance. Any larger number of quantiles will lead to a decrease of classification accuracy. Furthermore, with this choice of quantile number, the computational runtime is relatively low. This suggests that our method can achieve sufficient accuracy without requiring excessive computational overheads.

For directed graphs MPCA offers some immediate advantages since it is based on an explicit tensor representation of the data, it is convenient when dealing with high-dimensional histogram data which would normally need encoding in matrix or vector form. So although the performance advantages are fairly marginal for undirected graphs, for directed graphs it can offer significant performance advantages. However, this is at the expense of significantly increased computational cost.

In summary, based on these experimental observations we claim that our entropic histogramming and component analysis technique is both accurate and computationally efficient in classifying graphs extracted from real-world systems when the appropriate parameters and component analysis methods are selected. Moreover, the method is applicable to both undirected and directed graphs.

4. Conclusions

To conclude, in this paper we have suggested a novel and effective method that constructs entropy histograms over the edges of a network, and then performs ECA over a sample of such histograms in vector form for different networks. As a result each network is embedded as a point in a low-dimensional space spanned by the entropy components. In other words, for both undirected and directed networks we effect the embedding using the multivariate distribution of local von Neumann entropy with vertex degree combinations for the edges. This provides a complexity level characterization of graph structure based on the statistical information residing edge degree distribution.

Our analysis commences from an approximation of the von Neumann entropy, expressed in terms of the degrees of pairs of vertices connected by edges. The idea is to use the degree-based indexing of the edge entropy contributions to construct an entropy histogram. Such a distribution of edge-based entropy clearly encodes a number of intrinsic structural properties of a network, allowing us to obtain a simple entropic characterization of network structure. We have shown how such a histogram can further be encoded as a feature vector. In effect, by performing PCA or MPCA on the feature vectors we have proposed a network embedding method that embeds both undirected and directed graphs into a low-dimensional feature space, spanned by the entropy components of greatest variance.

We have experimented with a number of variants of this method on real-world network data. The variants arise through the choices of a) whether we used adaptive or fixed binning of the histogram, b) whether we use PCA or MPCA to perform variance component analysis over the set of histograms for the sample of graphs, and c) whether we histogram the raw number of edges of different vertex degree combination or whether we weight these combinations by their entropy. Our experiments have demonstrated that compared to a number of state-of-the-art network embedding methods, each variant of our method is more effective in terms of classification accuracy. Moreover, compared to kernel methods, the variant of our method with adaptive binning in conjunction with PCA is particularly computationally efficient and gives comparable classification performance. Focusing on the different available variants of our method, there are a number of conclusions that can be drawn. First, PCA is considerably more efficient than MPCA, without massive loss of classification accuracy. Second, when fixed full resolution binning of degree is used then there is no marked advantage of using entropy weighting over raw edge counts. On the other hand, when adaptive binning is used there appears to be an advantage in using entropy weighting.

Our method offers a number of advantages over existing methods in terms of how it captures vari-

ations in network structure. By performing entropy component analysis on the histogram of edge entropies indexed by the degree configurations of the edges, we identify those types of edges that have greatest variance in a sample of network. So if a network undergoes changes of structure, then our representation identifies those types of edges (in terms of degree composition) that are responsible for this change in structure, and moreover, allows us to identify those networks where the change is greatest as quantified by the change in entropy.

The work reported in this paper can be extended in a number of ways. First, it would be interesting to explore how the distribution of the edge entropy in a network can contribute to the development of novel information theoretic divergences, distance measures and relative entropies. Another interesting line of investigation would be to investigate whether this measure can be applied further to weighted graphs and hypergraphs. In the future, we also intend to explore novel and effective graph kernels defined over the inner products of our entropy distribution feature vectors.

REFERENCES

S. Battiston and G. Caldarelli. Systemic risk in financial networks. *Journal of Financial Managements Markets and Institutions*, 1:129–154, 2013.

C. H. Bennett. On the nature and origin of complexity in discrete, homogeneous, locally-interacting systems. *Foundations of Physics*, 16:585–592, 1986.

D. Berwanger, E. Gradel, L. Kaiser, and R. Rabinovich. Entanglement and the complexity of directed graphs. *Theoretical Computer Science*, 463:2–25, 2012.

G. Bonanno, G. Caldarelli, F. Lillo, S. Miccichè, N. Vandewalle, and R. N. Mantegna. Networks of equities in financial markets. *European Physical Journal B*, 38:363–372, 2004.

E. Z. Borzeshi, M. Piccardi, K. Riesen, and H. Bunke. Discriminative prototype selection methods for graph embedding. *Pattern Recognition*, 46:1648–1657, 2013.

G. Caldarelli, S. Battiston, D. Garlaschelli, and M. Catanzaro. Emergence of complexity in financial networks. *Lecture Notes in Physics*, 650:399–423, 2004.

F. Chung. Laplacians and the Cheeger inequailty for directed graphs. *Annals of Combinatorics*, 9:1–19, 2005.

J. C. Claussen. Offdiagonal complexity: A computationally quick complexity measure for graphs and networks. *Physica A*, 375:365–373, 2006.

M. Dehmer. Information processing in complex networks: Graph entropy and information functionals. *Applied Mathematics and Computation*, 201:82–94, 2008.

M. Dehmer, A. Mowshowitz, and F. Emmert-Streib. *Advances in Network Complexity.* Wiley-Blackwell, 2013.

F. Escolano, E. R. Hancock, and M. A. Lozano. Heat diffusion: Thermodynamic depth complexity of networks. *Physical Review E*, 85(036206), 2012.

E. Estrada. Quantifying network heterogeneity. *Physical Review E*, 82(066102), 2010.

D. Feldman and J. Crutchfield. Measures of statistical complexity: Why? *Physics Letters A*, 238:244–252, 1998.

L. Han, F. Escolano, E. R. Hancock, and R. C. Wilson. Graph characterizations from von Neumann entropy. *Pattern Recognition Letters*, 33:1958–1967, 2012.

R. Jenssen. Kernel entropy component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:847–860, 2010.

H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. *International Conference on Machine Learning*, pages 321–328, 2003.

A. N. Kolmogorov. On tables of random numbers. *Theoretical Computer Science*, 207:387–395, 1998.

J. Körner. Coding of an information source having ambiguous alphabet and the entropy of graphs. *6th Prague conference on information theory*, pages 411–425, 1973.

G. Li, M. Semerci, B. Yener, and M. J. Zaki. Effective graph classification based on topological and label attributes. *Statistical Analysis and Data Mining*, 5(4):265–283, 2012.

H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Mpca: Multilinear principal component analysis of tensor

objects. *IEEE Transactions on Neural Networks*, 19(1):18–39, 2008.

B. Luo, Wilson R. C., and E. R. Hancock. Spectral embedding of graphs. *Pattern Recognition*, 36:2213–2230, 2003.

A. K. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library (coil-20). *Technical Report*, February (CUCS-005-96), 1996.

F. Passerini and S. Severini. The von Neumann entropy of networks. *Inthernational Journal of Agent Technologies and Systems*, pages 58–67, 2008.

T. K. DM. Peron and F. A. Rodrigues. Collective behavior in financial markets. *Europhysics Letters*, 96(48004), 2011.

J. Platt. *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines.* MIT Press, 1999.

P. Ren, R. C. Wilson, and E. R. Hancock. Graph characterization via Ihara coefficients. *IEEE Transactions on Neural Networks*, 22:233–245, 2011.

I. Schomburg, A. Chang, C. Ebeling, M. Gremse, C. Heldt, G. Huhn, and D. Schomburg. Brenda, the enzyme database: updates and major new developments. *Nucleic Acids Research*, 32(D431-D433), 2004.

N. Shervashidze, S. V. N. Vishwanathan, T. H. Petri, K. Mehlhorn, and K. M. Borgwardt. Efficient graphlet kernels for large graph comparison. *12th International Conference on Artificial Intelligence and Statistics*, pages 488–495, 2009.

N. Shervashidze, P. Schweitzer, E. J. V. Leeuwen, and K. M. Borgwardt. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, 1:1–48, 2010.

F. N. Silva, C. H. Comin, T. K. DM. Peron, F. A. Rodrigues, C. Ye, R. C. Wilson, E. R. Hancock, and L. da F. Costa. On the modular dynamics of financial market networks. *ArXiv e-prints*, (1501.05040), 2015.

R. C. Wilson, E. R. Hancock, and B. Luo. Pattern vectors from algebraic graph theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1112–1124, 2005.

C. Ye, R. C. Wilson, C. H. Comin, L. da F. Costa, and E. R. Hancock. Approximate von Neumann entropy for directed graphs. *Physical Review E*, 89(052804), 2014.